

Universitatea POLITEHINICA București  
Facultatea de Automatică și Calculatoare

## Managementul Proiectelor Software

Hierarchy Analysis  
Software Development Document  
v 1.0

Echipa Farik

**București, 2012**

# Cuprins

Cuprins .....	2
1. Scopul documentului .....	3
2. Vedere de ansamblu a solutiei proiectate ...	3
TODO [ALL] Vom completa dupa finalizarea documentului	

# 1. Scopul documentului

Acest document are rolul de a descrie soluția tehnică proiectată pentru componenta *Hierarchy Analysis* din cadrul sistemului OCR modular pe care studenții îl dezvoltă în cadrul cursului de Managementul Proiectelor Software din cadrul Facultății de Automatică și Calculatoare.

Documentul servește drept ghid pentru implementarea soluției software pentru modulul *Hierarchy Analysis*.

## 2. Vedere de ansamblu a soluției proiectate

### 2.1. Fond

Funcțiile și scopul acestui proiect sunt redactate în Documentul Specificațiilor de Proiect la adresa <http://elf.cs.pub.ro/mps/wiki/proiect/descriere>.

### 2.2. Context și design de baza

TODO [ALL] Vom specifica doar dacă vom considera necesar care este flow-ul principal, eventual și o diagramă.

### 2.3. Decizii de implementare

TODO [DEV] Vom specifica care este limbajul folosit (java), pe ce platformă vom dezvolta, IDE-uri etc și în plus de ce s-a luat această decizie.  
Alte considerente: Java e free, fast, secure, reliable, cross platform etc.

### 2.4. Formatul fișierelor utilizate

Aplicație folosește fișiere XML ca modalitate de introducere a datelor în sistem. Datele de ieșire sunt tot fișiere XML.

Toate fișierele XML cu care lucrează componenta *Hierarchy Analysis* sunt validate de către XSD-urile puse la dispoziție la această adresă <http://elf.cs.pub.ro/mps/wiki/proiect/specificatii-xsd>.

## 4. Modelul datelor:

**HierarchyAnalysis** : scopul acestei clase este obținerea unui fișier XML care conține rezultatul analizei ierarhice. Îi permite utilizatorului să și aleagă modul în care va obține

fișierul de output XML. Prima variantă : încarcă unul sau mai multe fișiere XML obținute în urma analizei de layout și rulează un analizor ierarhic pentru acestea. În urma rularii va obține fișierul cu rezultatul analizei ierarhice. Cea de a doua posibilitate este încărcarea directă a fișierului de către utilizator și este cea recomandată de către ALINA!!!!!!

Structura clasei :

Membrii clasei sunt următorii:

- Un grup de elemente `RadioButton` care permite utilizatorului selecția unui anumit mod de introducere a datelor. Există 2 moduri posibile pentru a obține fișierul de intrare :
  1. Selecția mai multor fișiere XML de layout și a unui analizor ierarhic. Pentru a selecta fișierele XML se va folosi `JFileChooser`-ul : `BrowseXMLs`, iar pentru selecția analizorului ierarhic se va folosi `JFileChooser`-ul : `BrowseExec`.
  2. Selecția unui singur fișier XML care conține rezultatul analizei ierarhice cu ajutorul unui `JFileChooser` numit : `BrowseUniqueXML`.

Validarea fișierelor XML de intrare se face după încărcarea acestora. Operația este realizată în background, utilizatorul fiind înștiințat dacă operația a avut succes cu ajutorul unui `StatusBar`, în care mesajele vor fi : Validare în curs, Fișiere XML valide sau Fișiere XML invalide.

- Un `Jbutton` : `ShowHierarchy`. La apăsarea acestui buton se va închide fereastra curentă și se va deschide o nouă fereastră care va afișa ierarhia și va permite utilizatorului să corecteze eventualele erori.

Diagrama UI a clasei:

**Hierarchy Analysis Application**

☐ Incarc fișiere de layout si un analizor ierarhic

Multiple layout files... **Select files**

Hierarchy file... **Select file**

**Generate hierarchy file**

☐ Incarca un singur fișier XML cu rezultatul analizei ierarhice

XML hierarchy file... **Select file**

**Show hierarchy**

**Status Bar**

**Editor** : scopul acestei clase este afișarea diferitelor tipuri de blocuri, unirea lor , spargerea lor sau recatalogarea diferitelor blocuri.

Structura clasei :

Membrii clasei sunt urmatorii :

Clasa va contine doua Panel-uri. Cel din stanga va contine o suprafata reprezentand pagina, pe care se vor afisa blocurile de text si tipul acestora. In dreapta fiecarui bloc se vor afla 2 ComboBox-uri, unul pentru a permite editarea respectivului bloc : split, merge, schimbare tip si celalalt pentru schimbarea fontului blocului.

Panel-ul din dreapta va contine o suprafata reprezentand pagina, pe care se vor afisa imaginile ce au fost prelucrate de catre analizatorul de layout. Daca sunt inspectate mai multe imagini , acestea vor putea fi afisate una deasupra celelalte in panel ul acesta. Imaginile sunt afisate in cadrul unui JScrollPanel. De asemenea, exista 2 butoane care permit zoom-ul pe o anumita pagina. Pentru cazul in care zoom-ul este prea mare, ea va putea fi inspectata cu ajutorul unui JScrollBar orizontal.

In centrul ecranului se va afla un JscrollBar care va derula in mod sincron cele doua emisfere. In cazul unei desincronizari pe verticala, exista in partea dreapta un alt JscrollBar prin care se va face potrivirea.

Blocurile din emisfera stanga sunt instante ale unei clase parinte Element. Blocurile pot fi de tipul: Paragraph, Title, Subtitle, Article, Body, PageNumber, Image. Caracteristicile comune tuturor blocurilor sunt: existenta unei instante a clasei Cadru ( vizibila in interfata grafica printr un poligon incadrator ) si continutul propriu-zis de informatie.

TODO [QA]

## Planul de testare

TODO [ALL]

## Cerinte de sistem

procesor, RAM, runtime envs, Internet connection speed etc